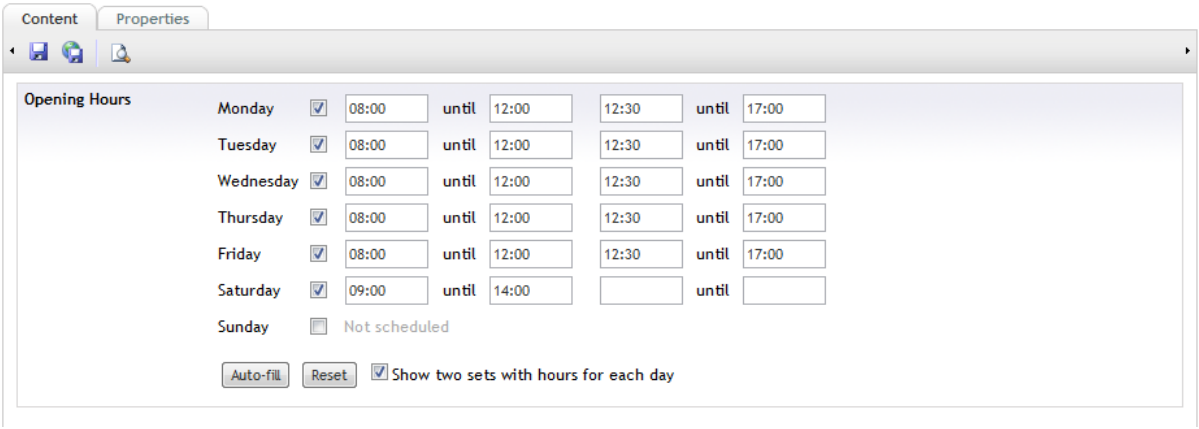
Project:	**OpeningHours**
Category:	**Data type / Backoffice extension**
Author:	**Lennart Stoop**
Date:	**25/06/2011**
Version:	**1.1.0.0**

OpeningHours - Installation and user guide

Table of contents

Introduction

This package contains a specialized data type for working with opening hours / office hours. The latest version can be downloaded freely from the Umbraco Deli and is provided under the MIT license: http://our.umbraco.org/projects/backoffice-extensions/openinghours



Requirements

- Umbraco version 4.7.0 or higher
- .NET version 3.5 or 4.0

Features

- Advanced data editor (ajax, auto complete, user-friendly)
- Supports 24-hour clock and 12-hour clock format (AM/PM)
- Various data editor settings allowing you to fine tune the data editor
- Localization support (adjusts to user language)
- Includes XSLT and Razor script examples

Change Log

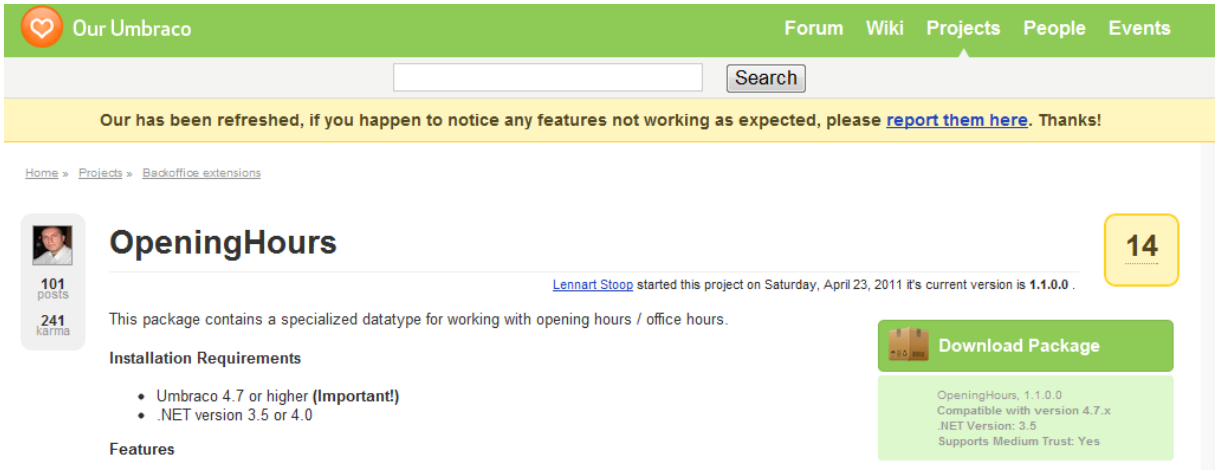
Version	Date	Description
1.0.0.0	23/04/2011	Initial release
1.0.0.1	25/04/2011	Jquery compatibility fix
1.0.0.2	28/04/2011	Bugfixes (Jquery, schedule validation, XSLT template)
1.1.0.0	24/06/2011	Support for 12-hour clock, abbrevation & localization of weekday names

Installation

Installation is quite easy and only takes a couple of minutes. In addition to this step by step guide you can also watch a short video demonstration on screencast.com: http://www.screencast.com/t/ZooBj9EWv9Oe

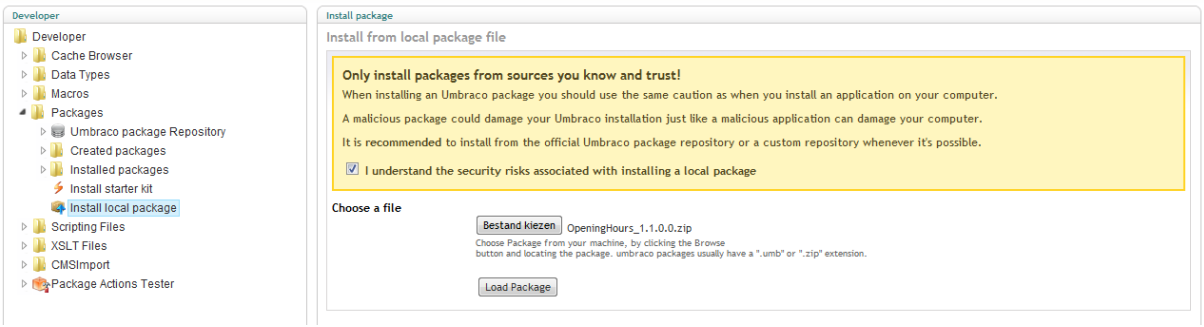
Step 1: Download the package from the Deli

Project URL: http://our.umbraco.org/projects/backoffice-extensions/openinghours

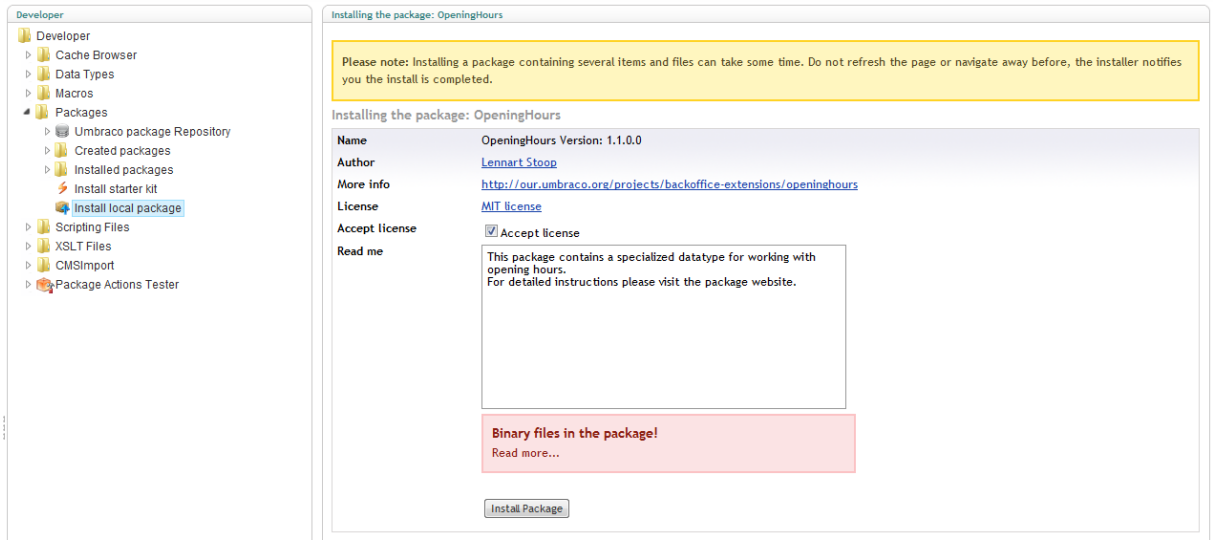


Step 2: Install the package in the Umbraco back office

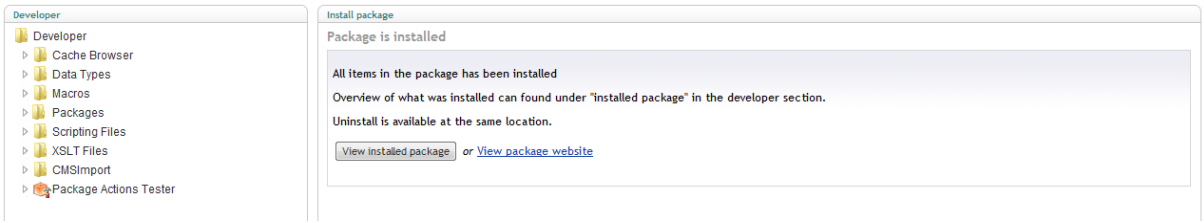
In the *Developer* section, expand the *Packages* folder, and select *install local package*. Select the package archive you just downloaded and check the installation agreement. Click the *Load Package* button to continue.



The next screen displays the package information. Check the *Accept license* checkbox and click the *Install Package* button to start the installation.



Wait for the progress bar to complete, and a final page appears that should confirm the success of the installation.



If at any given moment the installation fails, please don't hesitate to contact us through the forums: http://our.umbraco.org/projects/backoffice-extensions/openinghours/bugs

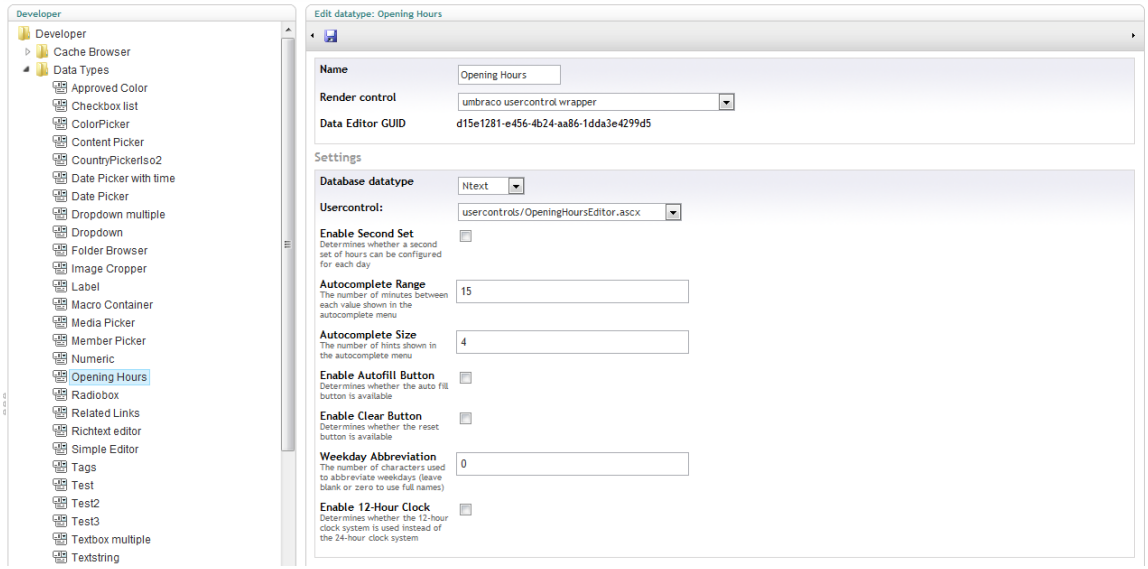
Components

After successful installation several files have been added to your Umbraco installation:

- The OpeningHours data type
- The OpeningHours macro
- Sample XSLT files
- Sample Razor scripts
- Resource files

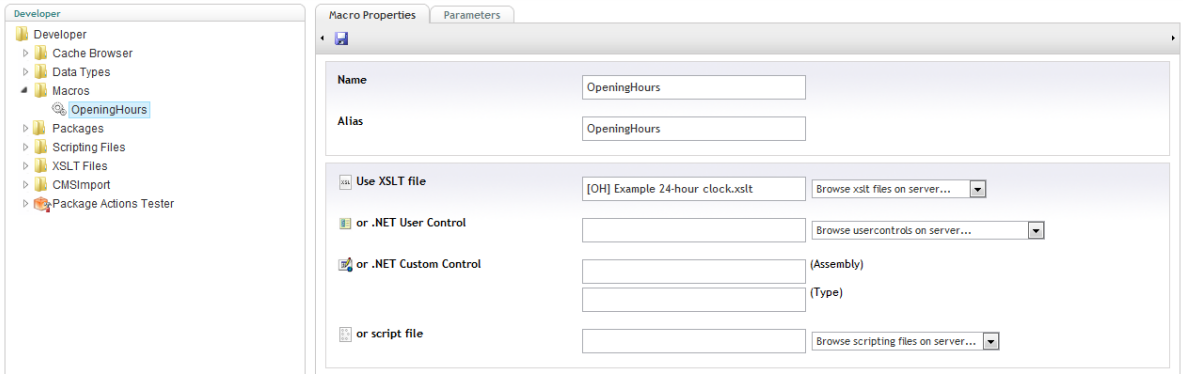
The OpeningHours data type

The data type can be found in the *Developer* section, by expanding the *Data Types* node and selecting the *OpeningHours* node. The data editor settings allow you to configure the data editor to your needs.



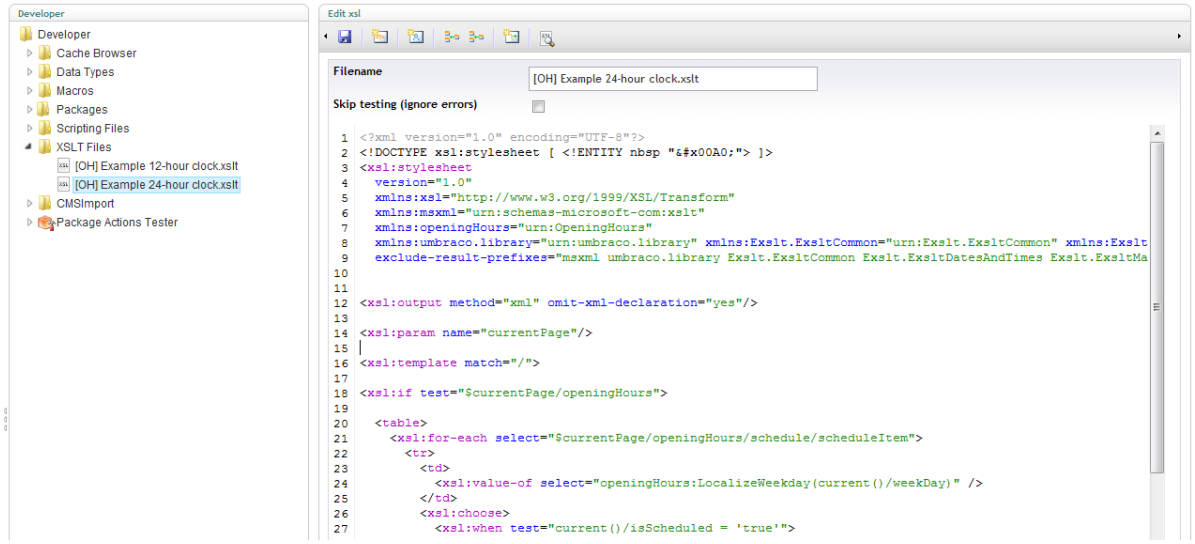
The OpeningHours macro

A macro can be found in the *Developer* section, by expanding the *Macros* node and selecting the *OpeningHours* node. The macro is preconfigured to use the example 24-hour clock XSLT file for content rendering, but if you prefer to use Razor or another script you can choose to configure one here.



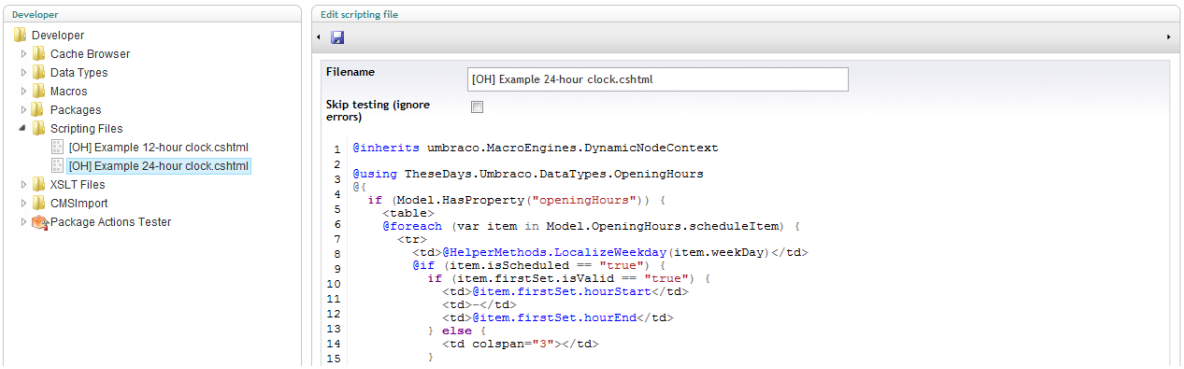
Sample XSLT files

Two sample XSLT files can be found in the *Developer* section, by expanding the *XSLT Files* node. These XSLT files allow you to get started really quickly, and show you how you can retrieve the XML data that is stored within the data type.



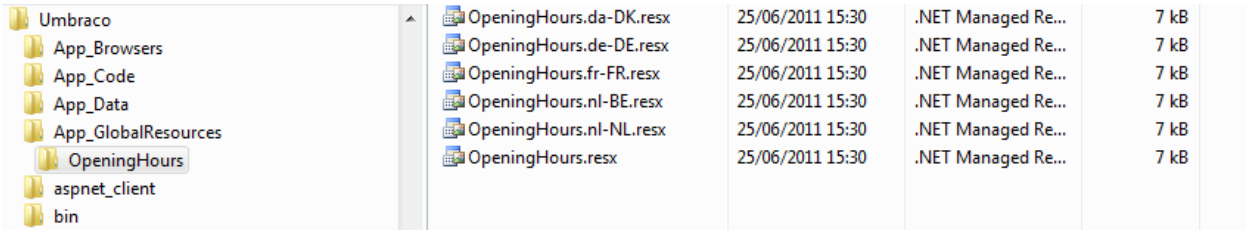
Sample Razor scripts

Two sample Razor scripts can be found in the *Developer* section, by expanding the *Scripting Files* node. The scripts render the exact same output as the XSLT files, and show you how you can retrieve the XML data that is stored within the data type.



Resource files

Several resources files can be found on the file system, in the *OpeningHours* folder within the *App_GlobalResources* folder. Resource files can be added or edited to localize the data editor used in the Umbraco back office (localization is done based on the user language of the active user which is configured in the *Users* section).

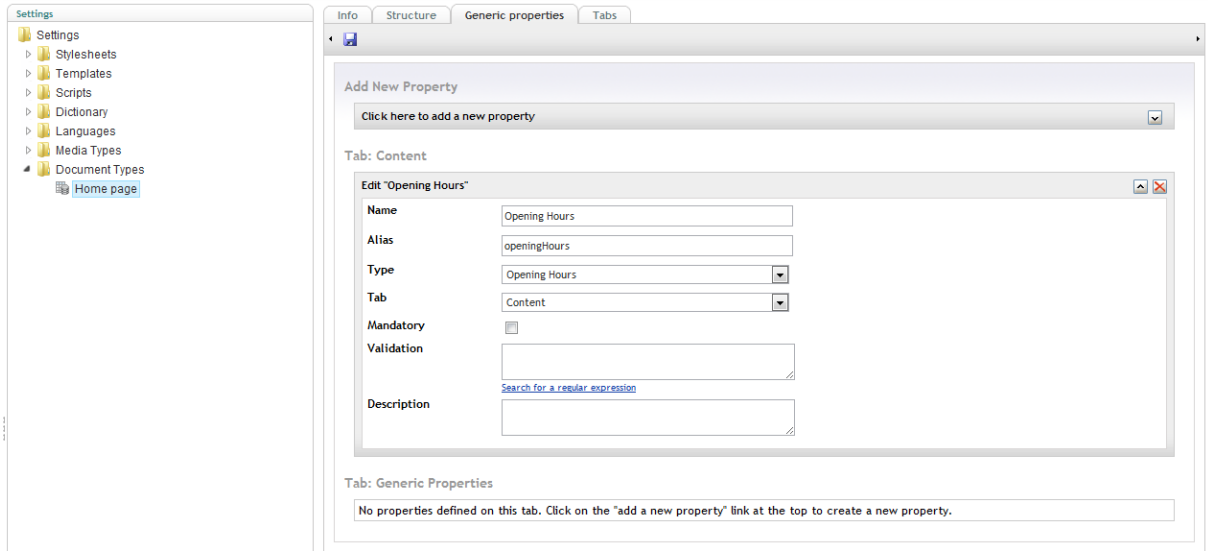


If your language is not supported and you need to create a new resource file, please do send us the resource file and we will make sure it is included in the next release.

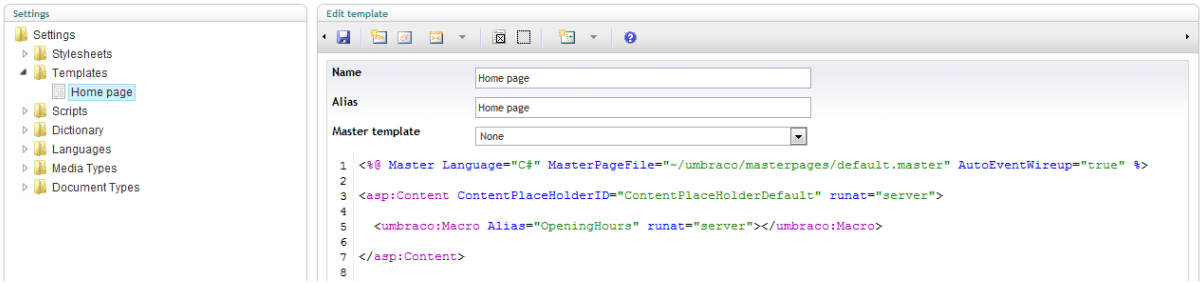
Quickstart

Once the package has been installed you can immediately start using the data type within your document types. If you are not familiar with document types or templates make sure you are by watching the free introduction movies on Umbraco TV: http://umbraco.com/help-and-support/video-tutorials/introduction-to-umbraco

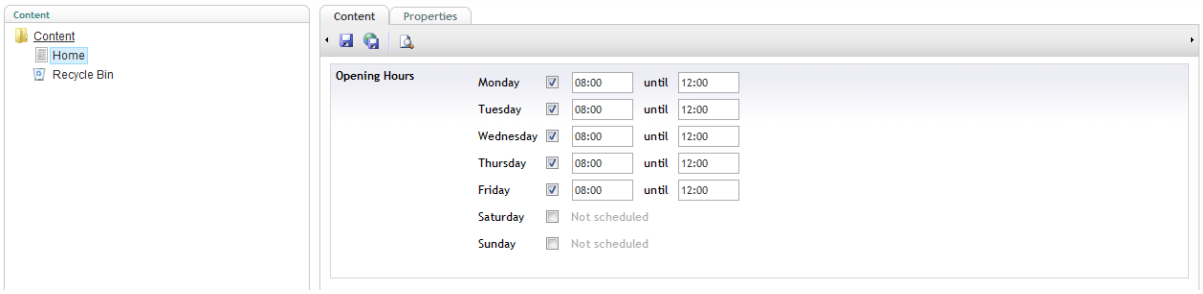
1) Create a new document type and add the *OpeningHours* data type as a property.



2) Add the *OpeningHours* macro to the template associated with the document type.



3) Create a document in the content tree, configure the opening hours schedule and publish the document.



4) Browse to the page in your front-end website and watch the content being rendered.

Data editor settings

A number of data editor settings allow you to configure the look and feel of the data editor. Configuration can be done in the *Developer* section, by expanding the *Data Types* node and selecting the *OpeningHours* node.

Settings overview

Setting	Description	Default
Enable Second Set	Determines whether a second set of hours can be configured for each day	False
Autocomplete Range	The number of minutes between each value shown in the autocomplete menu	15
Autocomplete Size	The number of hints shown in the autocomplete menu	4
Enable Autofill Button	Determines whether the auto fill button is available	False
Enable Clear Button	Determines whether the reset button is available	False
Weekday Abbreviation	The number of characters used to abbreviate weekdays (leave blank or zero to use full names)	0
Enable 12-Hour Clock	Determines whether the 12-hour clock system is used instead of the 24-hour clock system	False

Enabling 12-hour clock format

Enabling 12-hour clock format only affects the data editor UI. To ensure compatibility and allowing you to switch between formats, data is always stored in 24-hour clock format. If you wish to display the 12-hour clock format in the front-end, you will need to use the XSLT/Razor helper method which converts data to 12-hour clock format.

Data XML Format

Data is stored in a specific XML format of which a snippet is shown below. The sample XSLT and Razor scripts included in the package should give you an idea of how to access the XML data by iterating through the schedule items and evaluating the conditional settings.

```
    <schedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <hasSecondSet>true</hasSecondSet>
      <scheduleItem>
        <weekDay>Monday</weekDay>
        <isScheduled>true</isScheduled>
        <firstSet>
          <isValid>true</isValid>
          <hourStart>08:00</hourStart>
          <hourEnd>12:00</hourEnd>
        </firstSet>
        <secondSet>
          <isValid>true</isValid>
          <hourStart>12:30</hourStart>
          <hourEnd>17:00</hourEnd>
        </secondSet>
      </scheduleItem>
      <scheduleItem>
        <weekDay>Tuesday</weekDay>
        <isScheduled>true</isScheduled>
        <firstSet>
          <isValid>true</isValid>
          <hourStart>08:00</hourStart>
          <hourEnd>12:00</hourEnd>
        </firstSet>
        <secondSet>
          <isValid>true</isValid>
          <hourStart>12:30</hourStart>
          <hourEnd>17:00</hourEnd>
        </secondSet>
      </scheduleItem>
    </schedule>
```

XSLT and Razor helper methods

The data type provides 2 helper methods that can be used in both XSLT and Razor scripts.

Method	Description
LocalizeWeekday(String weekday)	Localizes the name of the weekday. In a multilingual website with languages configured through “manage hostnames” the configured language is used. If no language is configured the visitor’s browser language is used.
ConvertTo12HourClock(String hour)	Converts 24-hour clock format into 12-hour clock format, for example: 17:00 → 5:00 PM

Examples can be found in the sample XSLT and razor scripts. The snippets below show you how to use the methods in XSLT and Razor respectively:

```
<xsl:stylesheet
      version="1.0"
      xmlns:openingHours="urn:OpeningHours">

<xsl:value-of select="openingHours:LocalizeWeekday(current()/weekDay)" />
```

```
@using TheseDays.Umbraco.DataTypes.OpeningHours
@{
   @foreach (var item in Model.OpeningHours.scheduleItem) {
      @HelperMethods.LocalizeWeekday(item.weekDay)
   }
}
```